

(19) 日本国特許庁 (J P)

(12) 公開特許公報 (A)

(11) 特許出願公開番号

特開平7-249060

(43) 公開日 平成7年(1995)9月26日

(51) Int.Cl.<sup>6</sup>

G 0 6 F 17/50

識別記号

庁内整理番号

F I

技術表示箇所

7623-5L

G 0 6 F 15/ 60

3 6 0 D

審査請求 未請求 請求項の数 3 F D (全 9 頁)

(21) 出願番号

特願平6-65632

(22) 出願日

平成6年(1994)3月9日

(71) 出願人 000004237

日本電気株式会社

東京都港区芝五丁目7番1号

(72) 発明者 長谷川 拓己

東京都港区芝五丁目7番1号 日本電気株式会社内

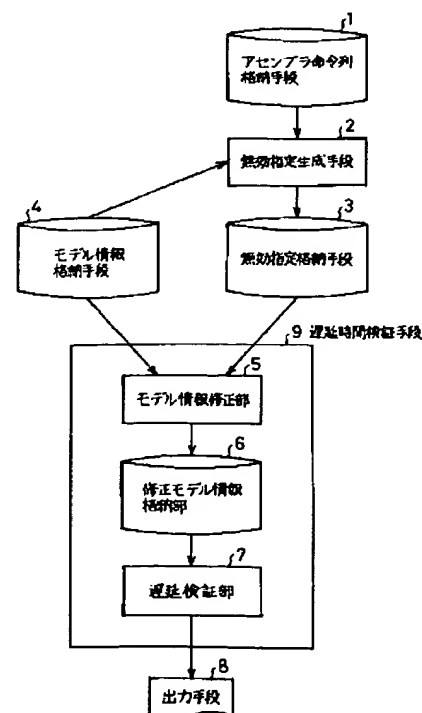
(74) 代理人 弁理士 境 廣巳

(54) 【発明の名称】 論理回路の遅延時間検証装置

(57) 【要約】

【目的】 論理回路に入力予定の全てのアセンブラ命令列を考慮して、その論理回路の遅延検証を実際の使用条件の下で行えるようにする。

【構成】 アセンブラ命令列格納手段1は検証対象となる論理回路に印加することが予定されている全てのアセンブラ命令列を保持する。無効指定生成手段2はこれらの全てのアセンブラ命令列に基づき、各アセンブラ命令列には存在しないビットの変化パターンを、各ビット位置毎に求め、その結果に従って、立ち上がり或いは立ち下がりの変化パターンが存在しないビットが印加される論理回路の端子を始点ノードとするアークの立ち上がり或いは立ち下がり種別に関する無効指定を生成して無効指定格納手段3に出力する。遅延時間検証手段9は無効指定格納手段3に格納された無効指定に従ってモデル情報格納手段4に格納されたモデル情報を修正し、この修正後のモデル情報に基づいて論理回路の遅延時間の検証を行う。



Best Available Copy

## 【特許請求の範囲】

【請求項 1】 遅延時間の検証対象となる論理回路の各外部端子および各回路素子の端子をノードとし、各ノード間の信号の流れをその方向として有するアークとした有向グラフ情報と、これら各アークの重みとして各々が有する遅延時間を、アーク対応に始点ノードおよび終点ノードの立ち上がり／立ち下がり種別毎に保持する遅延時間情報とで構成されたモデル情報を格納するモデル情報格納手段と、

前記論理回路に印加することが予定されている、ビット表現のアセンブラ命令列の全てを、各ビットが印加される位置を特定する情報と共に格納するアセンブラ命令列格納手段と、

該アセンブラ命令列格納手段に格納された全てのアセンブラ命令列に基づき、各アセンブラ命令列には存在しないビットの変化パターンを、各ビット位置毎に求め、その結果に従って、立ち上がり或いは立ち下がりの変化パターンが存在しなかったビットが印加される前記論理回路の端子を始点ノードとするアークの立ち上がり或いは立ち下がり種別に関する無効指定を生成する無効指定生成手段と、

該無効指定生成手段で生成された無効指定を格納する無効指定格納手段と、

該無効指定格納手段に格納された無効指定と前記モデル情報格納手段に格納されたモデル情報に基づき、遅延時間の検証を行う遅延時間検証手段とを含むことを特徴とする論理回路の遅延時間検証装置。

【請求項 2】 前記遅延時間検証手段は、前記無効指定格納手段に格納された無効指定に従って前記モデル情報格納手段に格納されたモデル情報中の遅延時間情報を修正するモデル情報修正部と、該モデル情報修正部による修正後のモデル情報に基づいて前記論理回路の遅延時間の検証を行う遅延検証部とを含むことを特徴とする請求項 1 記載の論理回路の遅延時間検証装置。

【請求項 3】 前記モデル情報修正部は、前記無効指定で指示されたアークの立ち上がり或いは立ち下がり種別に対応する前記モデル情報中の遅延時間を無効に修正することを特徴とする請求項 2 記載の論理回路の遅延時間検証装置。

## 【発明の詳細な説明】

## 【0001】

【産業上の利用分野】 本発明は論理回路の遅延時間検証装置に関し、特に論理回路をグラフ理論に従ってモデル化し、このモデル情報に従って論理回路の遅延検証を行う装置に関する。

## 【0002】

【従来の技術】 従来のこの種の論理回路の遅延時間検証装置について、図 2 に示す簡単な論理回路を例にして説明する。

【0003】 図 2 の論理回路は、回路素子 20 と外部入力端子 21、22 と外部出力端子 23 とを含み、外部入力端子 21 に印加された信号名 A の 1 ビットの信号と外部入力端子 22 に印加された信号名 B の 1 ビットの信号とから回路素子 20 において信号名 C の 1 ビットの信号を生成し、それを外部出力端子 23 に出力する回路である。

【0004】 このような論理回路をグラフ理論に従ってグラフ化すると、図 3 の如く表記される有向グラフが得られる。即ち、外部入出力端子および回路素子の端子をノード p~u とし、各ノード間の信号の流れをその方向として有するアーク a~e とする有向グラフである。ここで、アーク a は外部入力端子 21 に相当するノード p を始点ノード、回路素子 20 の一方の入力端子に相当するノード q を終点ノードとするアーク、アーク b は外部入力端子 22 に相当するノード t を始点ノード、回路素子 20 の他方の入力端子に相当するノード u を終点ノードとするアーク、アーク c は回路素子 20 の出力端子に相当するノード r を始点ノード、外部出力端子 23 に相当するノード s を終点ノードとするアーク、アーク d とアーク e は回路素子 20 の内部信号経路にそれぞれ対応するアークで、アーク d の始点ノードは q、終点ノードは r、アーク e の始点ノードは u、終点ノードは r である。なお、各ノード p~u の情報中には自ノードに印加される信号名の情報が含まれる。

【0005】 更に、各アーク a~e の重みとして、図 4 に示すように各アークの遅延時間を示す遅延時間情報がグラフ情報の一部として与えられる。この遅延時間は各アークの始点ノードおよび終点ノードの立ち上がり

(R) または立ち下がり (F) の種別 (以下、R/F 種別とも称す) 毎に与えられるもので、該当する遅延時間が存在しない場合には、その場合の重みは未定義 (無効) 「×」とされる。ここで、或るアークの始点ノードと終点ノードの信号の立ち上がり (R) と立ち下がり (F) との組み合わせは、図 4 に示すように、始点が R、終点も R (R/R) の場合と、始点が R、終点が F (R/F) の場合と、始点が F、終点が R (F/R) の場合と、始点が F、終点も F (F/F) の場合の 4 通りの組み合わせが存在するが、図 2 の論理回路の例では、各アークの R/F、F/R 2 種の組み合わせ (インバータ機能の場合が考えられる) については考える必要がないので、未定義「×」とされており、他の 2 種 R/R、F/F の組み合わせについて、回路素子 20 の特性に従ってそれぞれ重みとして遅延時間が付与されている。なお、図 4 において、「0」、「1」、「2」は遅延時間 (単位は ns) である。

【0006】 実際の遅延検証時には、図 3、図 4 に示す情報がモデル情報として用意されており、a→d→c の第 1 のパスと、b→e→c の第 2 のパスとの各々について、図 4 の各アークの始点および終点の R/F 種別毎に

10

20

30

40

50

全ての重みが求められ、この求められた重みの和が各バスの遅延時間として求められる。そして、最後に同一の始終点ノードを持つ複数のバスについてその最大遅延時間が求められる。この遅延時間の検証を行う詳細な動作は後に詳述するが、図 3、図 4 に示した情報の場合、最大遅延時間として 2 ns が算出される。

#### 【0007】

【発明が解決しようとする課題】ところで、図 2 に例示した論理回路の実際の使用時には、この論理回路を使用者の意図した通りに動作させるためのアセンブラ命令列と呼ばれる命令列が当該論理回路に印加される。一般的には論理回路の外部入力端子 21、22 にアセンブラ命令列の各ビットが印加されるが、論理回路の種類によっては内部素子の端子に直接に印加される場合もある。以下の例では外部入力端子 21、22 に印加されるものとして説明する。

【0008】図 5 は図 2 の論理回路の外部入力端子 21、22 に印加することが予定されている全てのアセンブラ命令列の例を示しており、ここでは、3 つのアセンブラ命令列 51、52、53 を示している。各々のアセンブラ命令列 51～53 は、図 2 の論理回路の外部入力端子が 21、22 と 2 個なので、「00」、「01」、「10」、「11」という 4 種類の 2 ビットの命令（これをアセンブラ命令と呼ぶ）の何れかを 2 個以上並べて構成されている。即ち、アセンブラ命令列 51 は「11」、「10」、「01」の 3 つのアセンブラ命令を並べたもの、アセンブラ命令列 52 は「10」、「01」、「00」の 3 つのアセンブラ命令を並べたもの、アセンブラ命令列 53 は「01」、「00」の 2 つのアセンブラ命令を並べたものであり、各々、一連の動作により論理回路を使用者の意図した通りに動作させるためのものである。なお、各アセンブラ命令の上位ビットは信号名 A に対応付けられており、外部入力端子 21 に印加されるビットであることを、下位ビットは信号名 B に対応付けられており、外部入力端子 22 に印加されるビットであることを示している。

【0009】論理回路の遅延時間の検証は、その論理回路に入力される全てのアセンブラ命令列が決定している場合、これらのアセンブラ命令列の範囲内でタイミング的に問題となるバスを検出すれば足り、逆に任意のアセンブラ命令列の範囲で検証することは、実際の使用条件と相違するので好ましくない。このような観点で考察すると、図 2 の論理回路に適用されるアセンブラ命令列の全てが図 5 に示すものであった場合、信号名 A に対応するビットの変化パターンを調べると、各アセンブラ命令列が入力される直前の信号名 A の信号の状態は不定なので、アセンブラ命令列 51 については、不定→1→1→0 と変化し、アセンブラ命令列 52 については、不定 1→0→0 と変化し、アセンブラ命令列 53 については不定→0→0 と変化しており、何れのアセンブラ命令列に

おいても 0→1 という変化パターンは現れていない。このため、信号名 A については 0→1 という変化パターンが存在しないことを前提として遅延時間を検証する必要があるが、従来の論理回路の遅延時間検証装置は、そのような不存在の変化パターンを検出して遅延検証に反映する機構を具備していなかったため、実際の使用条件の下での検証結果が得られないという問題点があった。

【0010】本発明はこのような従来の問題点を解決したもので、その目的は、検証対象となる論理回路に適用することが予定されている全てのアセンブラ命令列に基づいて存在しないビットの変化パターンを検出し、そのような変化パターンが存在しないことを前提とした遅延検証を行うことができる論理回路の遅延時間検証装置を提供することにある。

【0011】なお、本発明者は先の特許出願（特願平 5-105017 号）において、信号の立ち上がり、立ち下りの一つのみが有効となるようなアークに関して、そのアークと無効となる立ち上がりまたは立ち下りの種別とを設計者自身が直接指定する無効指定手段と、この無効指定手段による指定情報に基づいてモデル情報を修正する修正手段と、この修正後のモデル情報に基づいて論理回路の遅延時間の検証を行う手段とを含む論理回路の遅延時間検証装置を提案している。このような既提案装置を使用すれば上記問題点は一応解消されるが、無効指定を設計者自身が作成しなければならないため、人手作成に起因する無効指定漏れや誤指定が生じる可能性がある。そこで、本発明の別の目的は、人的介入無しに無効指定の生成を可能とし、人手作成に起因する無効指定の指定漏れや誤指定を防止することにある。

#### 【0012】

【課題を解決するための手段】本発明は上記の目的を達成するために、遅延時間の検証対象となる論理回路の各外部端子および各回路素子の端子をノードとし、各ノード間の信号の流れをその方向として有するアークとした有向グラフ情報と、これら各アークの重みとして各々が有する遅延時間を、アーク対応に始点ノードおよび終点ノードの立ち上がり／立ち下り種別毎に保持する遅延時間情報とで構成されたモデル情報を格納するモデル情報格納手段と、前記論理回路に印加することが予定されている、ビット表現のアセンブラ命令列の全てを、各ビットが印加される位置を特定する情報と共に格納するアセンブラ命令列格納手段と、該アセンブラ命令列格納手段に格納された全てのアセンブラ命令列に基づき、各アセンブラ命令列には存在しないビットの変化パターンを、各ビット位置毎に求め、その結果に従って、立ち上がり或いは立ち下りの変化パターンが存在しなかったビットが印加される前記論理回路の端子を始点ノードとするアークの立ち上がり或いは立ち下り種別に関する無効指定を生成する無効指定生成手段と、該無効指定生成手段で生成された無効指定を格納する無効指定格納手

段と、該無効指定格納手段に格納された無効指定と前記モデル情報格納手段に格納されたモデル情報に基づき、遅延時間の検証を行う遅延時間検証手段とを有している。

#### 【0013】

【作用】本発明の論理回路の遅延時間検証装置においては、アセンブラ命令列格納手段が、遅延時間の検証対象となる論理回路の外部入力端子に印加することが予定されている全てのアセンブラ命令列を、各ビットが印加される位置を特定する情報（例えば信号名）と共に保持しており、遅延検証に際しては、無効指定生成手段が、アセンブラ命令列格納手段に格納された全てのアセンブラ命令列に基づき、各アセンブラ命令列には存在しないビットの変化パターンを、各ビット位置毎に求め、その結果に従って、立ち上がり或いは立ち下がりの変化パターンが存在しなかったビットが印加される前記論理回路の端子を始点ノードとするアークの立ち上がり或いは立ち下がり種別に関する無効指定を生成して無効指定格納手段に出力し、遅延時間検証手段が、無効指定格納手段に格納された無効指定とモデル情報格納手段に格納されたモデル情報に基づき、遅延時間の検証を行う。即ち、遅延時間検証手段は、無効指定格納手段に格納された無効指定に従ってモデル情報格納手段に格納されたモデル情報中の遅延時間情報を修正し、この修正後のモデル情報に基づいて論理回路の遅延時間の検証を行う。

#### 【0014】

【実施例】次に本発明の実施例について図面を参照して詳細に説明する。

【0015】図1を参照すると、本発明の一実施例の論理回路の遅延時間検証装置は、アセンブラ命令列格納手段1と、無効指定生成手段2と、無効指定格納手段3と、モデル情報格納手段4と、遅延時間検証手段9と、出力手段8とで構成されている。

【0016】モデル情報格納手段4は、遅延時間の検証対象となる論理回路のモデル情報を格納するもので、図2に示した論理回路の場合には、図3に示した有向グラフ情報および図4に示した遅延時間情報とで構成されるグラフ情報が予め格納されている。即ち、遅延時間の検証対象となる論理回路の各外部端子および回路素子の各端子をノードとし、各ノード間の信号の流れをその方向として有するアークとし、更にこれら各アークの重みとして各々が有する遅延時間を与え、且つこの遅延時間はアーク対応に始点ノードおよび終点ノードのR/F種別毎に与えられている。

【0017】アセンブラ命令列格納手段1は、モデル情報格納手段4に格納されたモデル情報で表現された論理回路に印加することが予定されている全てのアセンブラ命令列を、各ビットが印加される位置を特定する信号名と共に予め格納している。図5はアセンブラ命令列格納手段1に格納された全アセンブラ命令列を示しており、

3つのアセンブラ命令列51、52、53を含んでいる。各アセンブラ命令列51～53には、各ビットが入力される位置を指定する信号名A、Bが付加されている。

【0018】無効指定生成手段2は、アセンブラ命令列格納手段1に格納された全てのアセンブラ命令列に基づき、各アセンブラ命令列には存在しないビットの変化パターンを、各ビット位置毎に求め、その結果に従って、遅延時間検証手段9で利用できる形式の無効指定を生成し、無効指定格納手段3に出力する。

【0019】図6および図7は無効指定生成手段2の処理例を示すフローチャートであり、以下、無効指定生成手段2の動作を説明する。

【0020】無効指定生成手段2は、先ず、アセンブラ命令列格納手段1に格納されたアセンブラ命令列を参照して存在する全ての信号名を認識し、各信号名毎に立ち上がり検出フラグ、立ち下がり検出フラグを生成して、それら全てのフラグをオフに初期化する（S1）。図5の例では、信号名Aに対応する立ち上がり検出フラグ、立ち下がり検出フラグ、信号名Bに対応する立ち上がり検出フラグ、立ち下がり検出フラグの合計4個のフラグが生成され、それらが全てオフに初期化される。

【0021】次に1つ目の信号名である信号名Aに注目し（S2）、アセンブラ命令列格納手段1から1つ目のアセンブラ命令列51を入力する（S3）。そして、このアセンブラ命令列51中の1つ目のビット値「1」を変数Xに代入し（S4）、次のビット値「1」を変数Yに代入し（S5）、 $X - Y$ の演算を実行し（S7）、結果を判別する（S8）。

【0022】演算結果が負（-）であれば、 $0 \rightarrow 1$ の変化パターンを検出したことになるので、信号名Aに対応する立ち上がり検出フラグをオンにし（S9）、正

（+）であれば、 $1 \rightarrow 0$ の変化パターンを検出したことになるので、信号名Aに対応する立ち下がり検出フラグをオンにし（S10）、結果が0であればフラグの操作は行わない。今の場合、X、Yとも1なので、結果は0であり、XにYを代入し（S12）、次のビット値

「0」をYに代入して $X - Y$ の演算を再び行う（S5、S7）。この結果は正なので、信号名Aに対応する立ち下がり検出フラグをオンにする（S10）。今回入力したアセンブラ命令列51には次のビットは存在しないため、ステップS6からS13へ進んで次のアセンブラ命令列52を入力し、ステップS4に戻ってアセンブラ命令列51と同様の処理を繰り返す。このアセンブラ命令列52の信号名Aのビットは $1 \rightarrow 0 \rightarrow 0$ と変化し、 $0 \rightarrow 1$ の変化は存在しないので、アセンブラ命令列52の処理後においても信号名Aに対応する立ち上がり検出フラグはオフのままである。次にアセンブラ命令列53が同様に処理されるが、信号名Aのビットは $0 \rightarrow 0$ と変化するだけで、 $0 \rightarrow 1$ の変化は存在しないので、アセンブラ

命令列53の処理後においても信号名Aに対応する立ち上がり検出フラグはオフのままである。アセンブラ命令列53の処理後に次のアセンブラ命令列を入力しようとした時点でアセンブラ命令列の終了が識別されるため、信号名Aに関する処理は終了し、次の信号名Bに注目を移し(S15)、信号名Bについて信号名Aと同様の処理を行う。

【0023】信号名Bについては、最初のアセンブラ命令列51における信号名Bの1番目のビット値「1」と2番目のビット値「0」との演算結果が正となるため、先ず信号名Bに対応する立ち下がり検出フラグがオンにされ、次に2番目のビット値「0」と3番目のビット値「1」との演算結果が負となるため、信号名Bに対応する立ち上がり検出フラグがオンにされる。この時点で双方のフラグが共にオンになるため、信号名Bに関する処理は終了し、ステップS11からステップS15、S16へ進み、ここで全ての信号名に注目し終えたことを判別し、図7の処理へと進む。

【0024】図7の処理においては、最初に信号名Aについて注目し(S21)、それに対応する立ち上がり検出フラグ、立ち下がり検出フラグの状態が判別される

(S22、S23)。そして、立ち上がり検出フラグがオフであれば、当該信号名Aの信号が入力される論理回路の端子をモデル情報格納手段4中の有向グラフにおけるノードの情報から特定し、この特定した端子を始点とするアークをモデル情報格納手段4中の遅延時間情報から特定し、この特定したアークの立ち上がり種別の重みを無効化する無効指定を生成して無効指定格納手段3に出力する(S24)。他方、立ち下がり検出フラグがオフであれば、当該信号名Aの信号が入力される論理回路の端子をモデル情報格納手段4中の有向グラフにおけるノードの情報から特定し、この特定した端子を始点とするアークをモデル情報格納手段4中の遅延時間情報から特定し、この特定したアークの立ち下がり種別の重みを無効化する無効指定を生成して無効指定格納手段3に出力する(S25)。今の場合、立ち上がり検出フラグがオフなので、図8に示すような無効指定がステップS24で生成されて無効指定格納手段3に出力されることになる。

【0025】信号名Aの次には信号名Bに注目を移して(S26)、信号名Aと同様の処理を行うが、今の場合はその立ち上がり検出フラグおよび立ち下がり検出フラグともオフでないため、信号名Aに関しては無効指定は生成されない。そして、ステップS27で全ての信号名に注目し終えたことを判別すると、無効指定生成処理を終了する。

【0026】さて、第1図を再び参照すると、遅延時間検証手段9は、無効指定格納手段3に格納された無効指定とモデル情報格納手段4に格納されたモデル情報に基づき、遅延時間の検証を行う手段であり、無効指定格納

手段3に格納された無効指定に従ってモデル情報格納手段4に格納されたモデル情報を修正するモデル情報修正部5と、その修正後のモデル情報を格納する修正モデル情報格納部6と、この修正後のモデル情報に基づいて論理回路の遅延時間の検証を行い、検証結果を出力手段8に出力する遅延検証部7とで構成されている。

【0027】図9はモデル情報修正部5の処理例を示すフローチャートである。モデル情報修正部5は、先ずモデル情報格納手段4から図2および図3に示したモデル情報を入力する(S30)。次に、無効指定格納手段3から1つ目の無効指定として、図8に示した無効指定を入力し(S31)、この無効指定に従ってモデル情報を修正する。即ち、無効指定で指定されたアークaの、無効指定で指定されたR/F種別であるR/Rの重みを無効「x」にする(S33)。そして、次の無効指定を無効指定格納手段3から入力し(S34)、存在すればその無効指定に従ってモデル情報を修正するが、今の場合、その他の無効指定は存在しないため、修正後のモデル情報を修正モデル情報格納部6へ出力し(S36)、モデル情報修正処理を終える。なお、無効指定格納手段3に1つも無効指定が存在しない場合には、ステップS32からステップS37へ進んで、入力したモデル情報そのものを修正モデル情報格納部6へ出力する。

【0028】以上のような処理がモデル情報修正部5で行われることにより、図4の情報は図10に示すように修正される。

【0029】図1の遅延検証部7は、修正モデル情報格納部6に格納された修正モデル情報に従って遅延検証を行うものであり、その動作は従来と同じである。以下、遅延検証部7の動作を図11のフローチャートに沿って説明する。

【0030】検証対象モデルの信号パスは複数存在するのが一般的であるので、各パス毎に遅延時間を算出し、最終的にこれら算出された遅延時間から、同一の始点および終点を持つパスを一つにまとめて最終的に遅延時間を算出する。そこで、先ず、縦形探索(深さ優先探索; Depth First Search, DFS)法により深さ方向のパスを求める(S41)。そして、このうちの1つのパスを第1のパスとして選択し(S42)、当該パスの全ノードのR/F種別を、例えばオールRにセットする(S44)。

【0031】図2の例では、第1のパスをa→d→cとすると、そのパスの各ノードはp, q, r, sであり、これら全てが信号立ち上がりRにセットされるものとする。そして、全ノードのR/F種別の現在の組み合わせ、すなわち今の場合はオールRにおける全アークa, d, cの重み(遅延時間)をそれぞれ求める(S46)。この重みは図10の修正後の情報から求められ、例えばアークaの重みについてみると、始点はp、終点はqであり、現在のそれらのR/F種別は全てRである

から、図10のアーカaの始点/終点のR/Rを参照して、重み「×」が検索される。他のアーカd, cの重みも同様に検索され、それらの結果が図12の最上行に示されている。

【0032】このようにして、全ノードのR/F種別がオールRの場合は、図12の最上行に示された各アーカの重みを得られ、それらの合計が算出される(S47)。但し、この場合の重みの合計処理においては、重みに1つでも無効「×」があれば、合計も無効「×」とされる。

【0033】次に、ノードの1つのR/F種別をFとして、2つ目のR/F種別の組み合わせとする(S48)。即ち、図12の2行目に示されるように各ノードR/Fの種別をオールR(R, R, R, R)から(F, R, R, R)の組み合わせとする。そして、このR/F種別の組み合わせでステップS46, 47を実行し、各アーカの重みを合計を求め、再びステップS48で、他のノードのR/F種別をFとして3つ目のR/F種別の組み合わせとする。即ち、図12の3行目に示される如く、各ノードのR/F種別を(R, F, R, R)の組み合わせとする。そして、このR/F種別の組み合わせでステップS46, S48を実行する。

【0034】このパスにはノードが4個存在するので、全ノードのR/F種別の組み合わせは $2^4$ 個あり、よってこの $2^4$ 個のR/F種別の組み合わせで、ステップS46, S47を実行する。最後の組み合わせであるオールF(F, F, F, F)の場合の各アーカの重みとその合計とが図12の最下行に示されている。

【0035】全ノードのR/F種別の全ての組み合わせ( $2^4$ の全て)が終了すると(S45でYES)、次の第2のパスが選択される(S49)。この第2のパスについても、上述の最初のパスと同様にステップS44, S46, S47, S48が順次繰り返し実行される。

【0036】この第2のパスについての実行結果が図13に示されている。なお、図13では、無効「×」を含む場合の結果は全て省略してある。

【0037】全てのパスについて上記動作が終了すると(S43でYES)、最後に、同一始点、終点ノードを持つ複数パスについて、最大遅延時間を持つ1つのパスにまとめる(S50)。

【0038】図12, 図13の例は、両パスが1つにまとめられるパスである。この場合、第1のパスの最大遅延時間は1nsであり、第2のパスの最大遅延時間も1nsであるため、最終結果は1nsとなる。この最終結果は図1の出力手段8に出力される。

【0039】これに対して、修正前の図4の情報をを用い

た遅延検証では、第1のパスの最大遅延時間が2nsとなるため、使用予定のアセンブラ命令列の範囲内での正確な検証が行えないことになる。

【0040】

【発明の効果】以上説明したように、本発明によれば、検証対象となる論理回路に適用することが予定されている全てのアセンブラ命令列に基づいて存在しないビットの変化パターンを検出し、そのような変化パターンが存在しないことを前提として遅延検証を行うため、論理回路の検証を実際の使用条件の下で実施することができ、また、非存在なビットの変化パターンの検出と、この検出結果に基づく無効指定の生成とを自動化したことにより、設計者の負担が軽減すると共に人的ミスの混入が防止できる。

【図面の簡単な説明】

【図1】本発明の一実施例のブロック図である。

【図2】検証対象とする論理回路の一例を示す図である。

【図3】図2の論理回路をグラフ理論に従って表現した有向グラフ図である。

【図4】図3の有向グラフにおける各アーカに対してR/F種別毎に付与される遅延時間の例を示す図である。

【図5】アセンブラ命令列の例を示す図である。

【図6】無効指定生成手段の処理例の一部を示すフローチャートである。

【図7】無効指定生成手段の処理例の残りの部分を示すフローチャートである。

【図8】無効指定の例を示す図である。

【図9】モデル情報修正部の処理例を示すフローチャートである。

【図10】無効指定に従って修正されたモデル情報を示す図である。

【図11】遅延検証部の処理例を示すフローチャートである。

【図12】遅延検証結果の例を示す図である。

【図13】遅延検証結果の例を示す図である。

【符号の説明】

1…アセンブラ命令列格納手段

2…無効指定生成手段

3…無効指定格納手段

4…モデル情報格納手段

5…モデル情報修正部

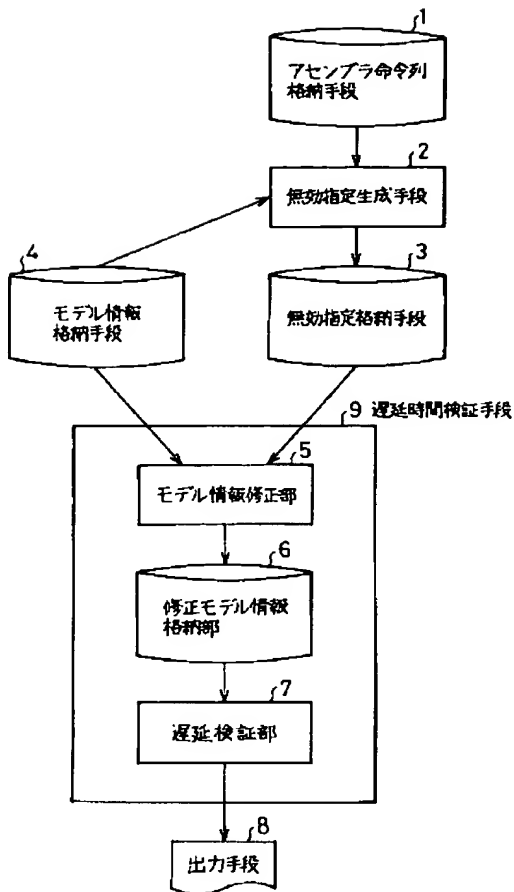
6…修正モデル情報格納部

7…遅延検証部

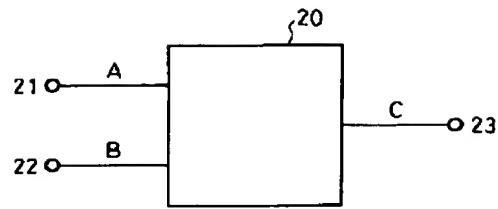
8…出力手段

9…遅延時間検証手段

【図 1】



【図 2】

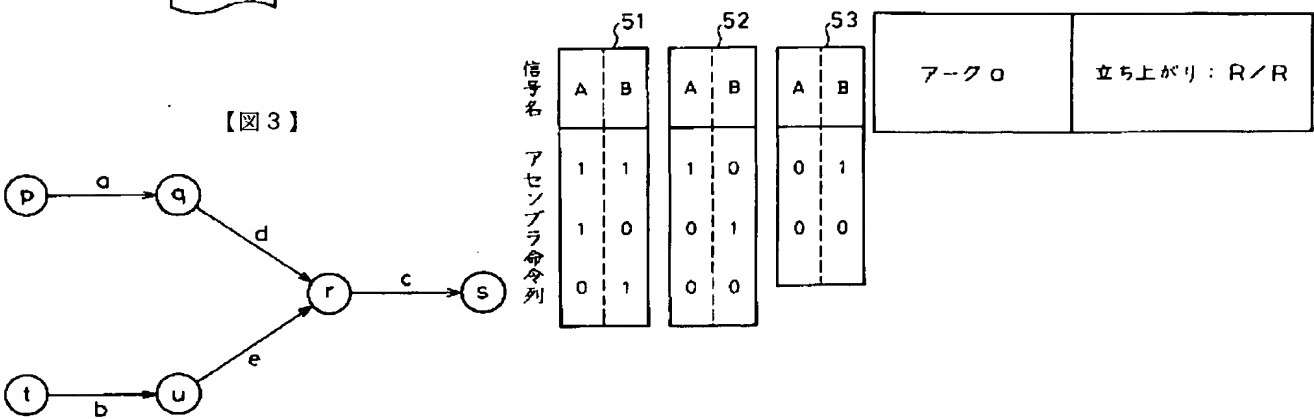


【図 4】

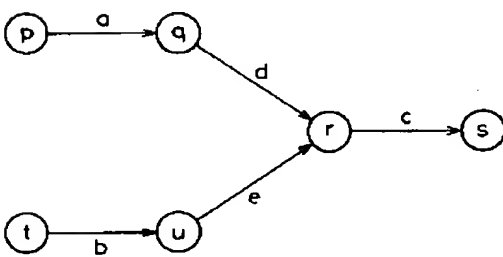
			始点 / 終点			
アーク	始点	終点	R/R	R/F	F/R	F/F
a	p	q	0	×	×	0
b	t	w	0	×	×	0
c	r	s	0	×	×	0
d	q	r	2	×	×	1
e	u	r	0	×	×	0

【図 5】

【図 8】



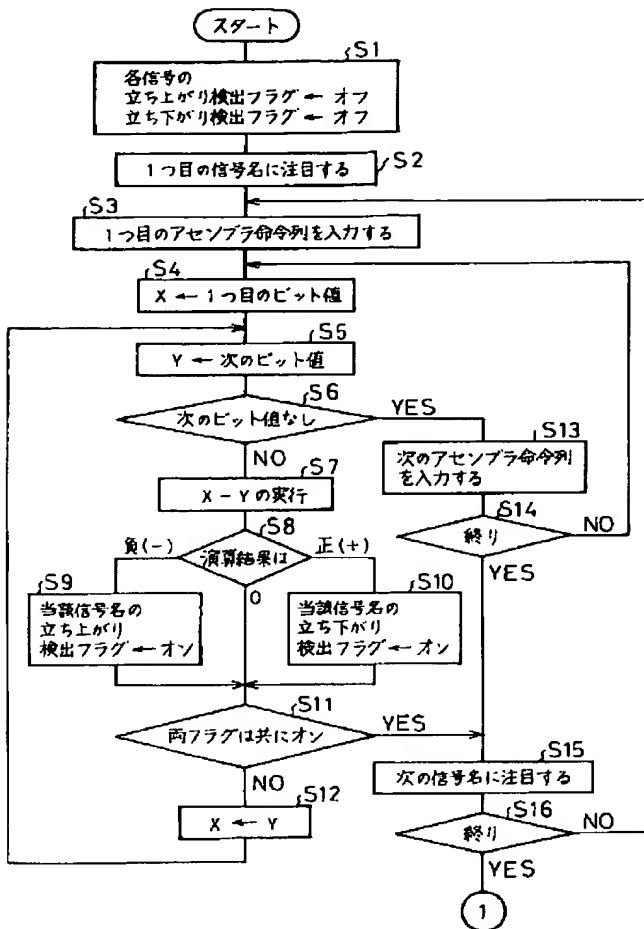
【図 3】



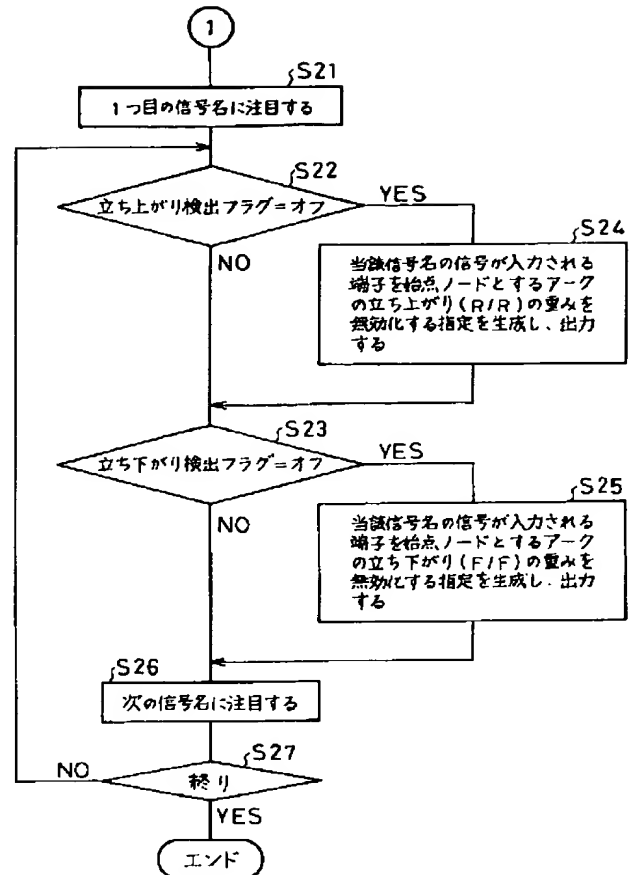
【図 13】

R/F 種別				遅延時間 T(ns)			
t	u	r	s	b	e	c	計
F	F	F	F	1	0	0	1.

【図 6】



【図 7】



【図 12】

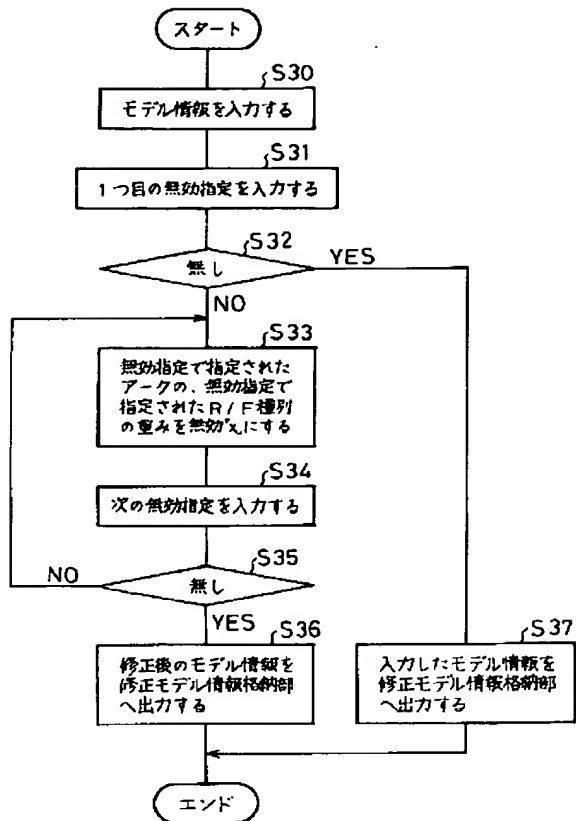
【図 10】

アーク	始点	終点	始点 / 終点			
			R/R	R/F	F/R	F/F
a	p	q	x	x	x	0
b	t	w	0	x	x	0
c	r	s	0	x	x	0
d	q	r	2	x	x	1
e	u	r	0	x	x	0

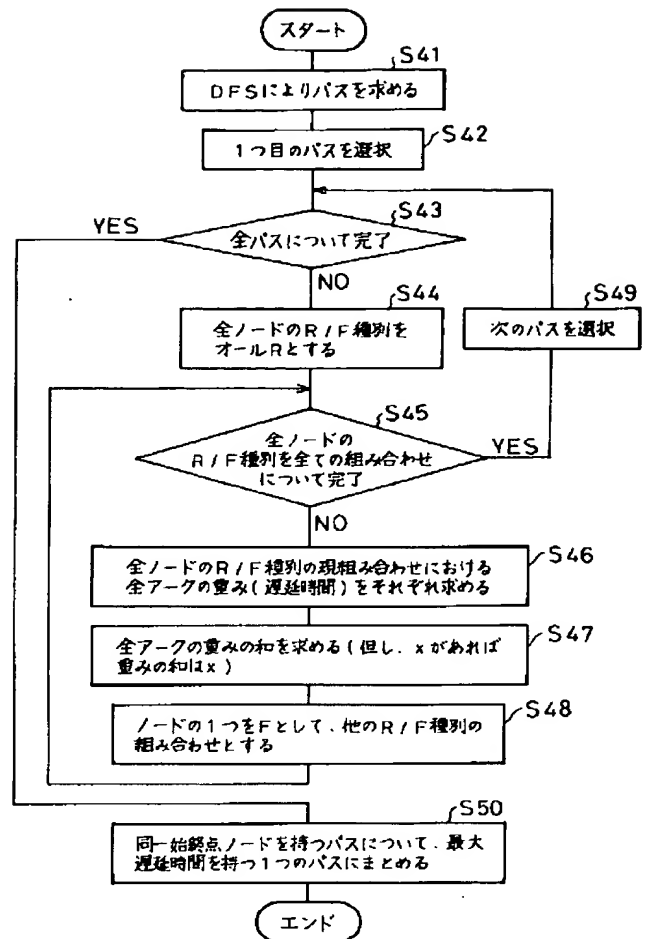
R/F 種別				遅延時間 T(ns)			
p	q	r	s	a	d	c	計
R	R	R	R	x	2	0	x
F	R	R	R	x	2	0	x
R	F	R	R	x	x	0	x
⋮	⋮	⋮	⋮	⋮	⋮	⋮	⋮
R	F	F	F	x	1	0	x
F	F	F	F	0	1	0	1



【図9】



【図11】



**This Page is Inserted by IFW Indexing and Scanning  
Operations and is not part of the Official Record**

## **BEST AVAILABLE IMAGES**

Defective images within this document are accurate representations of the original documents submitted by the applicant.

Defects in the images include but are not limited to the items checked:

- ☐ **BLACK BORDERS**
- ☐ **IMAGE CUT OFF AT TOP, BOTTOM OR SIDES**
- ☐ **FADED TEXT OR DRAWING**
- ☐ **BLURRED OR ILLEGIBLE TEXT OR DRAWING**
- ☐ **SKEWED/SLANTED IMAGES**
- ☐ **COLOR OR BLACK AND WHITE PHOTOGRAPHS**
- ☐ **GRAY SCALE DOCUMENTS**
- ☐ **LINES OR MARKS ON ORIGINAL DOCUMENT**
- ☐ **REFERENCE(S) OR EXHIBIT(S) SUBMITTED ARE POOR QUALITY**
- ☐ **OTHER: \_\_\_\_\_**

**IMAGES ARE BEST AVAILABLE COPY.**

**As rescanning these documents will not correct the image problems checked, please do not report these problems to the IFW Image Problem Mailbox.**